PASSWORD GENERATOR USING TKINTER

A MINI PROJECT REPORT

18CSC207J - ADVANCED PROGRAMMING PRACTICE

Submitted by

Aditi [RA2111003010314]
Palaash Surana[RA2111003010319]
Isha Singh[RA2111003010327]
Kamya Ojha[RA2111003010343]

Under the guidance of

Dr.A.M.J Muthu Kumaran

Assistant Professor, Department of Computer Science and Engineering

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled **"PASSWORD GENERATOR"** is the bona fide work of **Aditi [RA2111003010314], Isha Singh [RA2111003010327], Kamya Ojha [RA2111003010314] and Palaash Surana [RA2111003010319]** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

17/5/23

Dr.AMJ Muthu Kumaran
Assistant Professor
Department of Computing Technologies

SIGNATURE

Department of Computing Technology
Kattankulathur
603 203
SRM IST

M. Pushpalata

Dr. Pushpalata M
HEAD OF THE DEPARTMENT
Professor & Head
Department of Computing Technologies

SIGNATURE

Table Contents

ABSTRACT

A password generator is a computer program that generates unique and random passwords based on specific criteria. Python is a popular programming language that can be used to create a password generator with a Graphical User Interface (GUI).

The password generator in Python using GUI utilizes the Tkinter library, which is a standard GUI toolkit for Python. The program creates a user interface that allows users to set the length of the password they want and click on a generate button to generate a new password.

The password generator utilizes Python's built-in string module to generate random passwords. It includes a mix of uppercase and lowercase letters, digits, and special characters to create unique and strong passwords.

The password generator program is user-friendly and efficient, making it easy to generate strong passwords for multiple accounts. It provides an effective solution to the problem of creating strong and unique passwords that are difficult for hackers to crack.

INTRODUCTION

In today's digital age, it is more important than ever to have secure passwords for online accounts. Strong passwords are a critical line of defences against identity theft, data breaches, and other types of cyberattacks. However, creating strong passwords that are difficult for others to guess or crack can be a challenging task for many users.

To make the process of generating strong passwords easier and more convenient for users, a password generator using a graphical user interface (GUI) can be developed. A GUI-based password generator allows users to generate strong, randomized passwords for use in securing their online accounts with just a few clicks.
The GUI provides a user-friendly interface that makes the process of generating strong passwords quick and easy, even for users who are not technically savvy.

By using a Python framework such as Tkinter, developers can create a user-friendly interface that allows users to specify the desired length of the password and generate a strong and randomized password. The password generator function can be designed to use a combination of uppercase and lowercase letters, numbers, and special characters to generate strong passwords. With the increasing importance of online security, developing tools like password generators using GUI can help to make the online experience more secure and convenient for users.

METHODOLOGY

To develop a password generator using a GUI, the following methodology can be followed:

Plan and design the GUI: Identify the GUI elements required for the password generator, such as labels, input fields, buttons, and output fields. Plan the layout of the GUI and determine how the elements will be arranged on the screen.

Code the GUI using a Python framework: Use a Python framework such as Tkinter to code the GUI elements identified in the previous step. This involves creating the necessary widgets and arranging them on the screen.

Write the password generation function: Write a function that generates strong, randomized passwords. The function should take in the desired length of the password as an input and use a combination of uppercase and lowercase letters, numbers, and special characters to generate the password.

Integrate the password generation function with the GUI: Connect the password generation function to the "Generate" button on the GUI. When the user clicks the button, the function should be called, and the generated password should be displayed in the output field.

Test the password generator: Test the password generator to ensure that it works as expected. Try generating passwords of different lengths and verify that they are strong and randomized.

Refine the GUI and function: Based on user feedback and testing, refine the GUI and function as needed to improve usability and generate stronger passwords.

By following this methodology, a password generator using a GUI can be developed that provides users with an easy and convenient way to generate strong passwords for securing their online accounts.

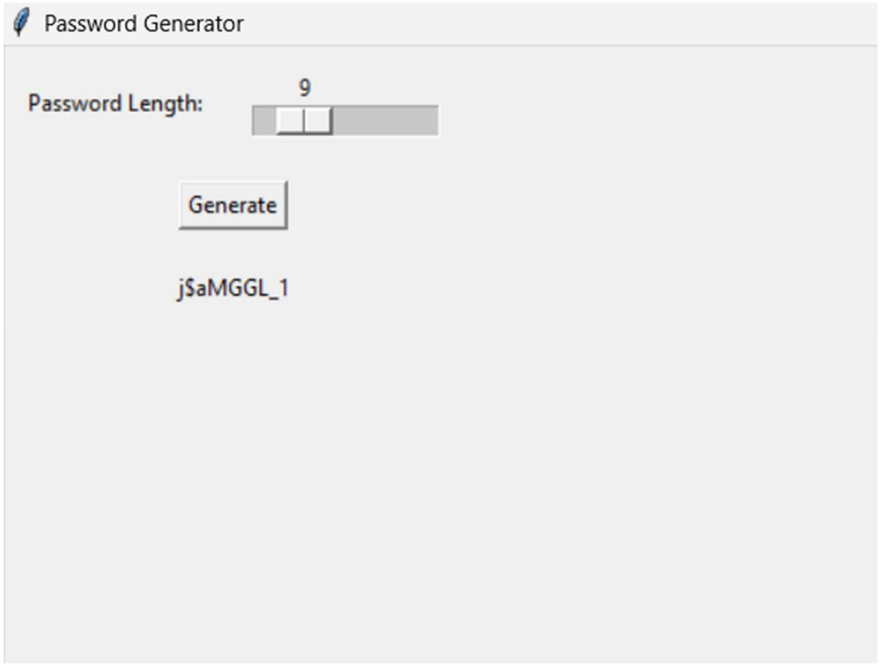
CODING AND TESTING

```
import random
import string
import tkinter as tk

def generate_password(length):
    """Generate a random password with the given length."""
    characters = string.ascii_letters + string.digits + string.punctuation
    password = ''.join(random.choice(characters) for i in range(length))
    return password

def generate():
    """Generate a new password and update the label."""
    password = generate_password(length_slider.get())
    password_label.config(text=password)

root = tk.Tk()
root.title("Password Generator")


length_label = tk.Label(root, text="Password Length:")
length_label.grid(row=0, column=0, padx=10, pady=10)
length_slider = tk.Scale(root, from_=4, to=32, orient="horizontal")
length_slider.grid(row=0, column=1, padx=10, pady=10)


generate_button = tk.Button(root, text="Generate", command=generate)
generate_button.grid(row=1, column=0, columnspan=2, padx=10, pady=10)


password_label = tk.Label(root, text="")
password_label.grid(row=2, column=0, columnspan=2, padx=10, pady=10)


root.mainloop()
```

OUTPUT AND RESULTS

Results:- Password Generator was successfully done and submitted. Screenshots of the output has been attached

Output:-



CONCLUSION AND FUTURE ENHANCEMENT

In conclusion, a password generator using a GUI can be a valuable tool for improving online security and making it easier for users to generate strong and randomized passwords. By using a Python framework such as Tkinter, developers can create a user-friendly interface that allows users to specify the desired length of the password and generate a strong and randomized password with just a few clicks.

There are several ways in which the password generator can be enhanced in the future. Some of these include:

Adding additional customization options: In addition to specifying the length of the password, users could be given the option to specify the types of characters they want to include, such as uppercase letters only, numbers only, or a combination of letters, numbers, and special characters.

Integrating with password management tools: Password generators could be integrated with password management tools to make it easier for users to securely store and manage their passwords.

Improving the password generation algorithm: The password generation algorithm could be improved to generate even stronger and more randomized passwords that are harder to crack.

Adding a password strength meter: A password strength meter could be added to the GUI to provide users with feedback on the strength of the generated password and suggestions for making it stronger.

Overall, a password generator using a GUI is a useful tool for improving online security and can be enhanced in many ways to provide even greater benefits to users.

REFERENCES

1. Python's official documentation: https://docs.python.org/3/tutorial/index.html
2. TkDocs: http://www.tkdocs.com/tutorial/index.html
3. Real Python: https://realpython.com/
4. Python for Beginners: https://www.pythonforbeginners.com/
5. Programiz: https://www.programiz.com/python-programming
6. tkinter 8.5 reference: https://tkdocs.com/tutorial/index.html
7. Tkinter (Example): https://www.python-course.eu/python_tkinter.php